

Restricción y Ordenación de Datos

Objetivos

Al finalizar esta lección, debería estar capacitado para:

- **Limitar las filas recuperadas por una consulta**
- **Ordenar las filas recuperadas por una consulta**
- **Utilizar la sustitución ampersand en *iSQL*Plus* para restringir y ordenar la salida en el tiempo de ejecución**

Limitación de Filas mediante una Selección


EMPLOYEES

EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90
103	Hunold	IT_PROG	60
104	Ernst	IT_PROG	60
107	Lorentz	IT_PROG	60
124	Mourgos	ST_MAN	50

...

20 rows selected.

**“recuperar todos
los empleados del
departamento 90”**



EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90

Limitación de las Filas que se Seleccionan

- **Restrinja las filas que se devuelven mediante la cláusula WHERE:**

```
SELECT * | { [DISTINCT] column | expression [alias], ... }  
FROM table  
[WHERE condition(s)];
```

- **La cláusula WHERE sigue a la cláusula FROM.**

Uso de la Cláusula WHERE

```
SELECT employee_id, last_name, job_id, department_id  
FROM employees  
WHERE department_id = 90 ;
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90

Cadenas de Caracteres y Fechas

- Las cadenas de caracteres y los valores de fecha van entre comillas simples.
- Los valores de caracteres son sensibles a mayúsculas/minúsculas y los valores de fecha son sensible al formato.
- El formato de fecha por defecto es DD-MON-RR.

```
SELECT last_name, job_id, department_id  
FROM employees  
WHERE last_name = 'Whalen' ;
```


Condiciones de Comparación

Operador	Significado
=	Igual que
>	Mayor que
>=	Mayor o igual que
<	Menor que
<=	Menor o igual que
<>	Distinto de
BETWEEN ...AND...	Entre dos valores (ambos inclusive)
IN (set)	Se corresponde con cualquier valor de una lista
LIKE	Se corresponde con un patrón de caracteres
IS NULL	Es un valor nulo

Uso de Condiciones de Comparación

```
SELECT last_name, salary
FROM employees
WHERE salary <= 3000 ;
```

LAST_NAME	SALARY
Matos	2600
Vargas	2500

Uso de la Condición BETWEEN

Utilice la condición BETWEEN para mostrar filas basadas en un rango de valores:

```
SELECT last_name, salary
FROM employees
WHERE salary BETWEEN 2500 AND 3500 ;
```

Límite inferior Límite superior

LAST_NAME	SALARY
Rajs	3500
Davies	3100
Matos	2600
Vargas	2500

Uso de la Condición IN

Utilice la condición IN de miembro para comprobar los valores incluidos en una lista:

```
SELECT employee_id, last_name, salary, manager_id
FROM   employees
WHERE  manager_id IN (100, 101, 201);
```

EMPLOYEE_ID	LAST_NAME	SALARY	MANAGER_ID
202	Fay	6000	201
200	Whalen	4400	101
205	Higgins	12000	101
101	Kochhar	17000	100
102	De Haan	17000	100
124	Mourgos	5800	100
149	Zlotkey	10500	100
201	Hartstein	13000	100

8 rows selected.

Uso de la Condición LIKE

- Utilice la condición **LIKE** para realizar búsquedas con comodines de valores válidos de cadenas de búsqueda.
- Las condiciones de búsqueda pueden contener literales de números o caracteres:
 - % denota cero o muchos caracteres.
 - _ denota un carácter.

```
SELECT  first_name  
FROM    employees  
WHERE   first_name LIKE 'S%';
```

Uso de la Condición LIKE

- Puede combinar caracteres que se correspondan con un patrón:

```
SELECT last_name  
FROM employees  
WHERE last_name LIKE '_o%' ;
```

LAST_NAME
Kochhar
Lorentz
Mourgos

- Puede utilizar el identificador ESCAPE para buscar los símbolos % y _ reales.

Uso de las Condiciones NULL

Compruebe si hay valores nulos con el operador IS NULL.

```
SELECT last_name, manager_id  
FROM employees  
WHERE manager_id IS NULL ;
```

LAST_NAME	MANAGER_ID
King	

Condiciones Lógicas

Operador	Significado
AND	Devuelve TRUE si <i>ambas</i> condiciones componentes son verdaderas
OR	Devuelve TRUE si <i>alguna</i> de las condiciones componentes es verdadera
NOT	Devuelve TRUE si la condición siguiente es falsa

Uso del Operador AND

AND requiere que ambas condiciones sean verdaderas:

```
SELECT employee_id, last_name, job_id, salary
FROM   employees
WHERE  salary >=10000
AND    job_id LIKE '%MAN%';
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
149	Zlotkey	SA_MAN	10500
201	Hartstein	MK_MAN	13000

Uso del Operador OR

OR requiere que una de las condiciones sea verdadera:

```
SELECT employee_id, last_name, job_id, salary
FROM   employees
WHERE  salary >= 10000
OR     job_id LIKE '%MAN%' ;
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
100	King	AD_PRES	24000
101	Kochhar	AD_VP	17000
102	De Haan	AD_VP	17000
124	Mourgos	ST_MAN	5800
149	Zlotkey	SA_MAN	10500
174	Abel	SA_REP	11000
201	Hartstein	MK_MAN	13000
205	Higgins	AC_MGR	12000

8 rows selected.

Uso del Operador NOT

```
SELECT last_name, job_id
FROM employees
WHERE job_id
      NOT IN ('IT_PROG', 'ST_CLERK', 'SA_REP') ;
```

LAST_NAME	JOB_ID
King	AD_PRES
Kochhar	AD_VP
De Haan	AD_VP
Mourgos	ST_MAN
Zlotkey	SA_MAN
Whalen	AD_ASST
Hartstein	MK_MAN
Fay	MK_REP
Higgins	AC_MGR
Gietz	AC_ACCOUNT

10 rows selected.

Reglas de Prioridad

Operador	Significado
1	Operadores aritméticos
2	Operador de concatenación
3	Condiciones de comparación
4	IS [NOT] NULL, LIKE, [NOT] IN
5	[NOT] BETWEEN
6	Distinto de
7	Condición lógica NOT
8	Condición lógica AND
9	Condición lógica OR

Puede utilizar paréntesis para sustituir las reglas de prioridad.

Reglas de Prioridad

```
SELECT last_name, job_id, salary
FROM employees
WHERE job_id = 'SA_REP'
OR job_id = 'AD_PRES'
AND salary > 15000;
```

1

LAST_NAME	JOB_ID	SALARY
King	AD_PRES	24000
Abel	SA_REP	11000
Taylor	SA_REP	8600
Grant	SA_REP	7000

```
SELECT last_name, job_id, salary
FROM employees
WHERE (job_id = 'SA_REP'
OR job_id = 'AD_PRES')
AND salary > 15000;
```

2

LAST_NAME	JOB_ID	SALARY
King	AD_PRES	24000

ORACLE

Uso de la Cláusula ORDER BY

- Para ordenar las filas recuperadas, utilice la cláusula ORDER BY:
 - ASC: orden ascendente, por defecto
 - DESC: orden descendente
- La cláusula ORDER BY se coloca al final de la sentencia SELECT:

```
SELECT last_name, job_id, department_id, hire_date
FROM employees
ORDER BY hire date ;
```

LAST_NAME	JOB_ID	DEPARTMENT_ID	HIRE_DATE
King	AD_PRES	90	17-JUN-87
Whalen	AD_ASST	10	17-SEP-87
Kochhar	AD_VP	90	21-SEP-89
Hunold	IT_PROG	60	03-JAN-90
Ernst	IT_PROG	60	21-MAY-91

...

20 rows selected.

ORACLE

Ordenación

- Ordenación en orden descendente:

```
SELECT last_name, job_id, department_id, hire_date  
FROM employees  
ORDER BY hire_date DESC;
```

1

- Ordenación por alias de columna:

```
SELECT employee_id, last_name, salary*12 annsal  
FROM employees  
ORDER BY annsal ;
```

2

- Ordenación por varias columnas:

```
SELECT last_name, department_id, salary  
FROM employees  
ORDER BY department id, salary DESC;
```

3

Variables de Sustitución



ORACLE

Variables de Sustitución

- **Utilice las variables de sustitución de *iSQL**Plus para:**
 - Almacenar valores temporalmente con las sustituciones de ampersand simple (&) y ampersand doble (&&)
- **Utilice variables de sustitución para complementar:**
 - Condiciones WHERE
 - Cláusulas ORDER BY
 - Expresiones de columna
 - Nombres de tablas
 - Sentencias SELECT completas

Uso de la Variable de Sustitución &

Utilice una variable con un ampersand (&) como prefijo para pedir un valor al usuario:

```
SELECT employee_id, last_name, salary, department_id
FROM   employees
WHERE  employee_id = &employee_num ;
```

Connected as **ORA1@T6**

 **Input Required**

Enter value for employee_num:

Uso de la Variable de Sustitución &

ORACLE[®]
iSQL*Plus

Logout Preferences Help

Workspace History

Connected as ORA1@T6

i Input Required

Enter value for employee_num:

Cancel Continue

1 2

old 3: WHERE employee_id = &employee_num

new 3: WHERE employee_id = 101

EMPLOYEE_ID	LAST_NAME	SALARY	DEPARTMENT_ID
101	Kochhar	17000	90

ORACLE[®]

Valores de Carácter y de Fecha con Variables de Sustitución

Utilice comillas simples para los valores de fecha y de carácter:

```
SELECT last_name, department_id, salary*12
FROM employees
WHERE job_id = '&job title' ;
```

 Input Required

Cancel

Continue

Enter value for job_title:

LAST_NAME	DEPARTMENT_ID	SALARY*12
Hunold	60	108000
Ernst	60	72000
Lorentz	60	50400

ORACLE

Especificación de Nombres de Columna, Expresiones y Texto

```
SELECT employee_id, last_name, job_id, &column_name  
FROM employees  
WHERE &condition  
ORDER BY &order_column ;
```

Input Required

Cancel

Continue

Enter value for column_name:

Cancel

Continue

Enter value for condition:

Cancel

Continue

Enter value for order_column:

ORACLE

Uso de la Variable de Sustitución &&

Utilice el ampersand doble (&&) si desea volver a utilizar el valor de variable sin pedir al usuario que realice una acción cada vez:

```
SELECT    employee_id, last_name, job_id, &&column_name
FROM      employees
ORDER BY  &column_name ;
```



Input Required

Cancel

Continue

Enter value for column_name:

EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
200	Whalen	AD_ASST	10
201	Hartstein	MK_MAN	20

...

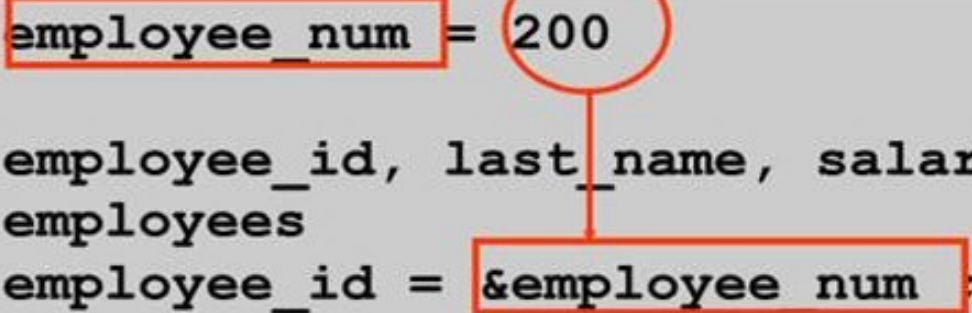
20 rows selected.

ORACLE

Uso del Comando DEFINE de *iSQL*Plus*

- Utilice el comando DEFINE de *iSQL*Plus* para crear y asignar un valor a una variable.
- Utilice el comando UNDEFINE de *iSQL*Plus* para eliminar una variable.

```
DEFINE employee_num = 200  
  
SELECT employee_id, last_name, salary, department_id  
FROM employees  
WHERE employee_id = &employee_num ;  
  
UNDEFINE employee_num
```

A diagram illustrating the use of the DEFINE command. A red box highlights 'employee_num' in the DEFINE statement, and another red box highlights '&employee_num' in the WHERE clause of the SELECT statement. A red circle highlights the value '200' in the DEFINE statement, with a red arrow pointing from it to the red box around '&employee_num'.

Uso del Comando VERIFY

Utilice el comando VERIFY para alternar la visualización de la variable de sustitución, tanto antes como después de que *iSQL**Plus reemplace con valores las variables de sustitución:

```
SET VERIFY ON  
SELECT employee_id, last_name, salary, department_id  
FROM employees  
WHERE employee_id = &employee_num;
```

"employee_num"

```
old      3: WHERE employee_id = &employee_num  
new      3: WHERE employee_id = 200
```

Práctica 2: Visión General

Esta práctica cubre los temas siguientes:

- **Selección de datos y cambio del orden de las filas que se muestran**
- **Restricción de filas mediante la cláusula WHERE**
- **Ordenación de filas mediante la cláusula ORDER BY**
- **Uso de variables de sustitución para agregar flexibilidad a las sentencias SELECT de SQL**